Jiaxuan

late TOken used 1, 1 remains

Go to https://learngitbranching.js.org. This is the best visual git explainer we know of.

1 Complete all the levels of main "Introduction Sequence". Report the commands needed to complete "Git rebase" with one line per command.

```
# git commit
# git checkout bugFix
# git rebase main
# git checkout main
# git commit
# git checkout bugFix
# git branch bugFix
```

2. Complete all the levels of main "Ramping up". Report the commands needed to complete "Reversing changes in git" with one line per command.

```
# git reset HEAD~1
# git checkout pushed
# git revert HEAD
```

3. Complete all the levels of remote “Push & Pull – Git Remotes!”. Report the commands needed to complete “Locked Main” with one line per command.

```
# git branch feature
# git checkout feature
# git push
# git checkout main
# git checkout feature
```

1 Exercise. For your pset submission, tell us only the answer to the last question (22).

```
# commit cd8f373050992529b870595560e263946279002b (HEAD -> master, origin/master, origin/HEA
# Author: Sergey Chekurin <schek.it@gmail.com>
# Date:   Wed Aug 28 19:37:18 2024 +1000
#
# Fix typo in 3-way-merge README.md
```

2. Exercise. For your pset submission, tell us only the output to the last question (18).

```
# type git diff in the terminal
# no output
```

#Merging # 1. Exercise. After completing all the steps (1 through 12), run git log –oneline –graph –all and report the output.

```
# Here are the outputs
# * cd8f373 (HEAD -> master, origin/master, origin/HEAD) Fix typo in 3-way-merge README.md
# * 7872289 Allow the use of older node
# *   5579118 (origin/JKrag/submodule-security-note) Merge branch 'master' into JKrag/submodu
# | |\
# | |/
# |/|
# * | c0ab7fb Add config file for CodeRabbit code review tool
# | * 14810d5 Update submodules/README.md
# | * b61c91c Update submodules/README.md
# | * 6425843 Add note to submodules kata about security workaround
# |/
# | * 7a310d6 (origin/jkrag/coderabbit) Add config file for CodeRabbit code review tool
# |/
# *   baeeb85 Merge pull request #361 from eficode-academy/wip/signed-commits
# |\
# | * 6a35030 (origin/wip/signed-commits) created initial GPG signing kata
# |/
# *   b59589c Merge pull request #359 from eficode-academy/jkrag/1000stars
# |\
# | * 1633c2f (origin/jkrag/1000stars) Celebrate 1000 Stars
# |/
# * 268f094 Add rebase-multiple-commits exercise
# | * 6c8eef8 (origin/jkrag/restore-add-sleep) Add sleep statments to setup
# |/
# * 273eeb0 Improve "ignore" kata
# | * c74d30b (origin/wip/enchance-readme) added sofus as maintainer
# |/
# | * 7f2b5dd (origin/uppercase) hertile
# | * c815298 Add an awesome new file
# |/
```

3. Identify the type of merge used in Q1 and Q2 of this exercise. In words, explain the difference between the two merge types, and describe scenarios where each type would be most appropriate.

```
# The first merge is a fast forward where the second is a three way
# In the first merge no new commits are done in the main branch; in the second, the master ar
```

Undo, Clean, and Ignore (10 points)

1. Exercise. Report the answer to step 13.

```
# path = "/Users/doudouliu/Desktop/python 2/student30538/problem_sets/git-katas"

# import os

# Define the path to the git-katas directory
# path = "/Users/doudouliu/Desktop/python 2/student30538/problem_sets/git-katas"

# List all files in the directory structure to confirm paths
# for root, dirs, files in os.walk(path):
#    for name in files:
#       print("File:", os.path.join(root, name))


# Update with the confirmed path from the list above
# file_path = "/Users/doudouliu/Desktop/python 2/student30538/problem_sets/git-katas/basic-cl

# Read and display the file content
# with open(file_path, "r") as file:
#    content = file.read()
#    print(content)
```

2. Exercise. Look up git clean since we haven't seen this before. For context, this example is about cleaning up compiled C code, but the same set of issues apply to random files generated by knitting a document or by compiling in Python. Report the terminal output from step 7.

```
# run the following in the terminal
# git clean -n -d    # Show files and directories that would be removed
# git clean -f -d    # Actually remove those files and directories
# git clean -f -x    # Remove all untracked and ignored files
# (no output)
```

#3. Exercise. Report the answer to 15 (“What does git status say?”)

```
# by running the git status
# doudouliu@mamemames-MacBook-Pro git-katas % git status
# On branch master
# Your branch is up to date with 'origin/master'.
```